

## 활 동 보 고 서

회차	6회차	지도교수 확인란	
일시	2019.9.4.(수) 19:00~22::00		
장소	충남대학교 도서관 그룹 스터디룸		
주제	Java(2) 비교문, 조건문, 반복문, 메소드, 객체지향 프로그래밍 등		
학습 내용 요약			
<div>&lt;메소드&gt;</div> <p>메소드(method)는 코드를 재사용할 수 있게 해준다.</p> <p>String은 이 메소드가 출력할 값이 무조건 문자열이라는 뜻</p> <p>자바가 위치한 파일로 들어가 <code>java -cp bin org.opentutorials.javatutorials.io InputDemo</code>를 입력하면 InputDemo파일이 실행.</p> <p>변수와 메소드로 이루어진 객체. 이 객체는 재사용이 가능하다. 일종의 부품으로써 사용가능.</p> <div>&lt;비교와 Boolean&gt;</div> <p>-Boolean</p> <p>불린(Boolean)은 참과 거짓을 의미하는 데이터 타입으로 bool이라고도 부른다. 불린은 정수나 문자와 같이 하나의 데이터 타입인데, 참을 의미하는 true와 거짓을 의미하는 false 두 가지의 값을 가지고 있다. 아래는 비교 연산자들에 대한 설명이다.</p> <div>&lt;조건문&gt;</div> <p>조건문은 if로 시작하고 if 뒤 괄호에는 조건을 쓴다.</p> <p>-if 문의 조건이 true 면 {부터 }까지 실행된다.</p> <p>-else는 if의 조건이 거짓일 때 else 부분을 실행시켜라라는 명령어이다.</p> <p>-else if는 조건문을 더 많이 쓸 수 있게 하는 명령어이다.</p> <p>-if 문 뒤 조건에 위 차트에서 배운 조건문을 결합해서 사용해도 된다.</p> <p>-&amp;&amp;는 and, 두 조건이 일치해야 하며</p> <p>  는 or, 두 조건 중 하나만 일치해도 된다는 뜻.</p> <p>조건문의 중첩</p> <p>조건문에서 사용빈도도 적고 if문으로 대체가능하지만, 때에따라서 보다 명료하게 로직을 보여줄 수 있는 switch문에 대해서도 배웠다.</p> <p>And는 좌항과 우항이 모두 참일 때 참인 식이고 &amp;&amp;라고 표현함. Or은 좌항과 우항 둘 중 하나만 참이면 참인 식이고   라고 표현한다. !은 not의 의미이고 boolean의 의미를 역전시킨다.</p>			

if다음의 구문이 한줄이면 중괄호를 생략해도된다.

### <반복문>

반복문에는 while가 for가 있는데

while 문은 while(조건){실행시킬 코드} 형식으로 쓰면 된다.

for 문은 for(변수 선언 초기값 지정; 조건; 증감식){실행시킬 코드}로 사용한다.

만약 반복문이 없다면 I like Teamlog를 100번 치고 싶다고 가정하면 document.write('I like teamlog');를 100번 입력해야 할 것이다.

반복을 하다가 중간에 중단시키고 싶으면 break;을 쓰면 된다.

비슷하지만 다른 continue 란 것이 있는데 continue는 그 부분을 중단시키고 반복은 계속 돌게 하는 명령어이다.

### <배열>

[] 기호를 사용(배열을 만들겠다는 기호)

대괄호 안에 데이터를 ,로 구분해서 나열하면 배열이 됨.

형식은 var 변수 이름 = ['데이터', '데이터', '데이터']

-> 이것을 입력했다면 변수 하나에 3개의 데이터를 담은 것이다.

각각의 데이터는 원소라고 부른다. 배열에 담겨 있는 값을 가져올 때는 대괄호 안에 숫자를 넣는데 0 부터 시작한다. 배열이 없다면 회원 정보를 만들겠다고 가정하면 직접 다 입력해야 하지만 배열을 사용하면 그냥 배열에다 배치시켜두고 load하면 되기 때문에 훨씬 간략하다는 느낌을 줄 수 있다.

For-each

for문을 이용해서 배열에 담긴 원소들을 하나씩 꺼내서 실행시켰는데, for each를 활용해 더 간결하게 표현할수있다.

### <객체>

배열과 객체는 비슷하다.

차이점은 데이터를 담아내는 그릇이라는 점에서는 비슷하지만 객체는 index의 값으로 어떤 숫자를 쓰는 것이 아니라 인덱스로 원하는 데이터를 지정할 수 있다는 점이다.

다른 언어에서는 연관 배열, 맵, 딕셔너리 등등의 데이터 타입이 객체에 해당하는 것이다.

배열은 대괄호로 시작해서 대괄호로 끝나지만 객체는 중괄호로 시작해서 중괄호로 끝난다.

코드 하나를 가져와서 설명하자면

```
{'kimseongi':100, 'jominseong':80, 'hwangjunyong':1}
```

이렇게 객체를 만들 수 있다.

저 숫자는 저 index에 해당하는 값이라고 할 수 있다. 이 객체를 사용하려면 저 객체를 대표하는 변수를 만들어줘야 한다. 변수에다가 객체를 담아내면 언제든지 사용할 수 있다.

```
var teamlog = {'kimseongi':100, 'jominseong':80, 'hwangjunyong':1};
```

이렇게 변수에다가 객체를 담아내고

`alert(teamlog['hwangjunyong']);`를 입력하면 결과는 1이 출력된다.

for 문은 in 뒤에 따라오는 배열의 key 값을 in 앞의 변수 name에 담아서 반복문을 실행한다. 객체에는 객체를 담을 수도 있고, 함수도 담을 수도 있다.

for (var hello in user) 이렇게 for 문을 사용하면  
value가 아니라 key 값을 0번째 배열부터 불러온다

#### <메소드>

메소드(method)는 코드를 재사용할 수 있게 해준다.

이것은 어떤 기능을 하는 부분을 말하는데 예를 든다면 핸드폰에서 소리를 늘리거나 크게 하기 위해 버튼을 누르면 핸드폰에 소리를 줄여라, 늘려라 하는 기능과 같다.

그렇다면 이

#### <객체지향 프로그래밍>

객체지향 프로그래밍(Object-Oriented Programming)은 좀 더 나은 프로그램을 만들기 위한 프로그래밍 패러다임으로 로직을 상태(state)와 행위(behavior)로 이루어진 객체로 만드는 것이다. 이 객체들을 마치 레고 블럭처럼 조립해서 하나의 프로그램을 만드는 것이 객체지향 프로그래밍이라고 할 수 있다. 다시 말해서 객체지향 프로그래밍은 객체를 만드는 것이다. 따라서 객체지향 프로그래밍의 시작은 객체란 무엇인가를 이해하는 것이라고 할 수 있다.

## 학습 방법 및 과정

6주차에는 학습 내용이 많기 때문에 각자 학습을 하기에다 벅찬 시간이었다. 따라서 남은 시간에 의견 교류를 할 시간이 부족해, 다른 주차에 비해 학습 방법 및 과정의 내용이 조금 적다.

입력값을 포함한 코드를 돌릴 때, 이클립스에서 실행시키려면 평소와는 다른 방식으로 실행해야한다. 기존의 방법대로 run을 하면 에러가 뜨기 때문에, run configuration에 들어가서 따로 입력값을 지정해줘야한다.

자바에 있어서 메소드는 중요한 역할을 하는데, 팀원들이 메소드가 무엇인지 감이 잘 안 온다고 해서, 그나마 이해를 한 신예슬 학우가 쉽게 설명을 해 주었다.

“우선 메소드를 만드는 이유는, 메소드의 역할에 있는데, 이것은 코드의 재사용이잖아요. 즉 만든 코드를 다른 부분에서 언제든지 호출해서 사용 할 수 있어요.”

메소드는 이렇게 `output(){ 내용 }` 의 형식으로 선언을 한다. `Public`과 `static void` 이 부분은 뒤에서 설명하고 일단 선언을 하고 `Main`. 메소드에서 `output();` 으로 호출해서 사용 할 수 있다

### <학습 소감>

신예슬 학우: 조건문, `boolean`, 반복문, 메소드 등의 모든 언어를 관통하는 기본적인 문법들은 파이썬을 먼저 배운 덕분인지 쉽게 이해가 갔다. 문법은 다르지만 기본원리와 쓰임새는 비슷했기 때문에 배우는데에 큰 어려움은 없었다. `if`문과 비슷하게 사용가능한 `switch`문에 대해 새로 배웠고, 여러 가지 문법들을 같이 코드에 적용해보면서 함께 사용할 때 어떻게 효율적인 코드를 짤 수 있는지 배웠다. 또, 항상 자바에서 새로운 클래스를 만들면 자동으로 입력되어있는 `public static void main(String[] arg){}`가 어떤 의미인지 정말 궁금했는데, 메소드를 배우면서 대략적으로 알 수 있어서 궁금증이 해결되었다.

강미규 학우: 파이썬도 잘 이해가 안 됐는데, 자바로 넘어오니 정말 이해가 안 돼서 혼났다. 다만 출력을 의미하는 `print`의 경우에는 파이썬과 루비, 그리고 자바의 모두가 `print`와 유사한 형태의 것?함수?를 사용하는 것에서 프로그래밍 언어는 모두 다르지만 어떤 점에서는 유사한 성질을 갖는다는 것을 어렵듯이 느꼈다. `if`문에서는 원인을 알 수 없는 에러가 발생해 한참을 붙잡고 있었는데, 알고보니 `cmd`입력창을 열어 파일의 경로값을 설정해주니 해결이 되어 허무하면서도 신기하기도 했다. 지난 번에도 적었듯이, 확실히 자바는 파이썬이나 루비보다 훨씬 예민하고 다루기 힘든 언어인 것 같다.

이정란 학우: 뭔가 java 식이 이해가 되는 것 같으면서 내가 혼자 해보려고 하면 반복문부터 확실히 헷갈린다. 힌트를 언제 안 보고 칠 수 있을 것인가! 그리고 요새 무기력증이 다시 도지고 있어서 체력의 한계로 인해 복습을 많이 하지는 못하였다. 공부도 체력이 뒤따라 줘야 오래 할 수 있기에 매주 4번 정도 저녁에 줄넘기를 하는 계획을 세웠다. 그리고 9월 달은 동아리 축제부스 운영과 강연준비로 많이 바쁠 것 같아 미리미리 끝내놓아야겠다는 생각도 들었다.

java 언어를 더 잘 이해하고 잘 복습해서 눈 감고도 코딩할 수 있는 날이 왔으면 좋겠다. 내일은 java 배열 수업 하나 듣고 최소 2번씩은 코드 다 손으로 직접 써봐야겠다.

김예린 학우: 다른 것들은 파이썬이나 루비에서도 배웠던 개념이라 조금 형태가 달라지긴 했어도 비슷한 느낌으로 이해하면 조금은 와닿는 느낌이었는데, 메소드라는 것이 정말 와닿지 않아서 한 두 번 정도 해당 회차 강의를 돌려보고, 예슬 학우에게도 물어보았다. 그럼에도 며칠 지나면 또 그 개념과 의미를 까먹을 것 같아서 걱정이 되기도 한다. 숲을 봐야 하는데 자꾸 이런 식의 나무를 봐서 코딩 실력이 안 느는 것 같다. 앞으로는 전반적인 코드의 흐름같은 것을 파악하려고 노력해야겠다.

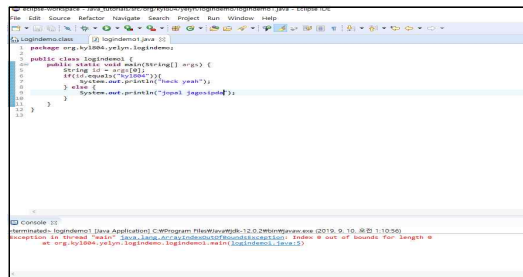
김현지 학우: 파이썬을 배우고나서 자바를 하니 그나마 나왔다. 그렇지만 자바는 신경써야 할 것이 많고 오류가 바로바로 뜨니까 긴장하면서 코딩을 하게 되는 느낌이 있다. 조건문이나 반복문은 해봤던 거라 이해하기 쉬웠다. 메소드나 객체지향프로그래밍은 파이썬루비에서 한 번 접해봤던 거라서 익숙하긴 했지만 그래도 아직은 수련이 많이 필요할 것 같다. 강의를 듣고 난 후 혼자서 생각하고 해봐야 할 것 같았다.

### 활동 사진 첨부

#### 학습 활동 ①

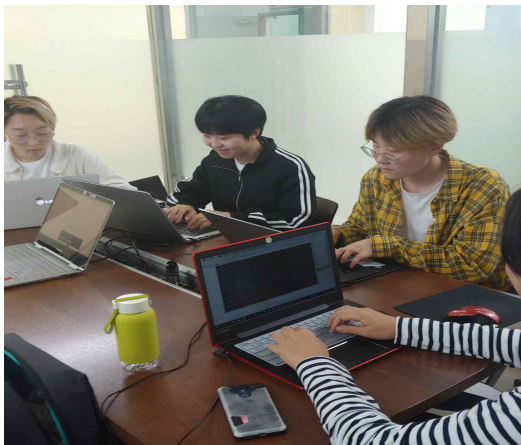


#### 학습 결과물 ①



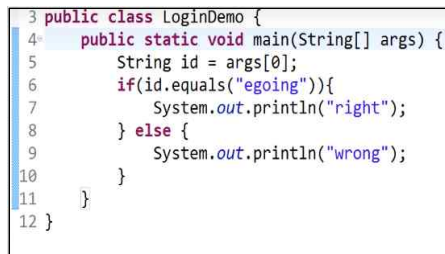
조건문을 실행했는데 오류가 난 화면이다.

#### 학습 활동 ②



(신예슬 학우가 찍은 사진 ㅎㅎ)

#### 학습 결과물 ②



함께 고민해봤지만 해결책을 찾지 못해 강의에 나온대로 코드를 작성했음.

> 문제 해결